Non-Provisional Patent Application

Attorney Docket No: **CSCO-012/4912**

DISTRIBUTION OF DATA TRANSFER LOAD WHEN TRANSMITTING LAYER-3 DATAGRAMS ON A LAYER-2 NETWORK

Inventors

Sudhakar SHENOY Second Floor 'B', Flat No.11 14th Cross, Sampige Road, Bangalore Karnataka (State), India - 560 003 Citizenship: India	Amit S. PHADNIS D-606, Pride Apartments Bannerghatta Road, Billekhalli Village Bangalore, India - 560 076 Citizenship: India

Assignee:

Cisco Technology, Inc.
A California Corporation
170 W. Tasman Drive
San Jose, CA 95134
Telephone: (408) 525-9706
Fax: (408) 526-5952

Attorney:

Law Firm of Naren Thappeta
Phone: +1 (510) 342-2519 x-6580
Fax: + 1 (707) 356-4172
Email: naren@iphorizons.com
URL: www.iphorizons.com

DISTRIBUTION OF DATA TRANSFER LOAD WHEN TRANSMITTING LAYER-3 DATAGRAMS ON A LAYER-2 NETWORK

Background of the Invention

Field of the Invention

The present invention relates to telecommunication networks, and more specifically to a method and apparatus for distributing data transfer load on different paths of a layer-2 network (e.g., ATM) when transporting layer-3 (e.g., Internet Protocol) datagrams.

Related Art

Layer-2 networks are often used to transport layer-3 datagrams. In a typical configuration well known in the relevant arts, an edge router interfaces with user systems (e.g., personal computers) and an asynchronous transfer mode (ATM) network to provide data transfers between the two. Another edge router may be present on the other side which provides data transfers between the ATM network and addition systems (e.g., target systems representing servers) accessed by the user systems.

A virtual circuit is often provided on a physical path (of a layer-2 network) to enable data transfers between edge routers. Thus, an edge router encapsulates a layer-3 datagram (received from a user system) in multiple ATM cells (or layer-2 packets, in general) and transmits the cells to the other edge router on the virtual circuit. The other edge router may forward the received data to a target system again in the form of layer-3 datagram(s). The data transfer in the reverse direction may also be performed similarly.

A virtual circuit may be shared for transferring data related to many systems. In general, it is

desirable to provide high bandwidths on virtual circuits such that the applications on end systems (and servers) are not impeded by bottlenecks (throughput and/or latency) in data transfers. At least to avoid such bottlenecks, it is often desirable to increase the data transfer capability between edge routers (or other end systems at which virtual circuits terminate).

In a known prior approach, a service provider may increase the bandwidth of the virtual circuit, for example, by employing faster underlying physical paths. Alternatively, the service provider may use a different path with higher bandwidth. Unfortunately such approaches increase the overall cost for implementing networks. In addition, the approaches typically lead to service disruption when increasing the bandwidth as the changes entail (re)configuration of edge routers and/or intermediate switches. Accordingly, such approaches may be undesirable at least in some environments.

In an alternative prior approach, multiple logical IP interfaces may be configured on each router, with each logical IP interface being assigned a single IP address. A virtual circuit may be provisioned between each pair of IP addresses, with one address on each edge router. As a result, two different IP (layer-3) routes would be deemed to be present between the two edge routers. The traffic load (IP datagrams) may be distributed among the different layer-3 routers. In other words, the load balancing is achieved at layer-3 level.

Due to such distribution, a high aggregate bandwidth may be available for data transfers between the edge routers. At the same time, additional costs to implement high bandwidth physical paths may be avoided. However, one problem with such an alternative approach is that the complexity of implementation may be enhanced due to the need to support multiple routes (routing table entries)

resulting from the parallel IP routes between the edge routers.

Accordingly, what is needed is a method and apparatus which enables data transfer load to be distributed on several virtual circuits provisioned potentially on different virtual circuit paths.

Summary of the Invention

An edge router in accordance with the present invention associates multiple layer-2 virtual circuits with a single layer-3 route. The virtual circuits may be provisioned on different physical paths of a layer-2 network. The edge router distributes the traffic load on the virtual circuits when transmitting the datagrams (destined on the layer-3 route).

As a result, a high aggregate data transfer capability may be provided for the layer-3 route without necessarily having to provide a correspondingly high bandwidth on a single underlying physical path (of the layer-2 network). In one embodiment, layer-3 corresponds to internet protocol (IP) and layer-3 corresponds to ATM such that IP datagrams are transferred on the ATM virtual circuits provisioned to another edge router.

In one implementation of an edge router, a forwarding block determines an IP route by accessing a forwarding table using a destination IP address. The forwarding table returns a route entry indicating an edge router at the next hop and an interface (on the edge router) on which the datagram is to be transmitted. A VC determination block then accesses a virtual circuit table to determine the specific one of the virtual circuits provisioned to the edge router at the next hop.

In an alternative implementation, a forwarding information base is implemented, which provides the most suited VPI/VCI when a lookup is performed based on the IP destination address. In an embodiment, the FIB is implemented in the form of a tree structure such that the tree needs to be traversed using a destination IP address as a key to determine the VPI/VCI.

Further features and advantages of the invention, as well as the structure and operation of various embodiments of the invention, are described in detail below with reference to the accompanying drawings. In the drawings, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number.

Brief Description of the Drawings

The present invention will be described with reference to the accompanying drawings, wherein:

Figure 1 is a block diagram illustrating an example environment in which the present invention can be implemented;

Figure 2 is a flow chart illustrating a method according to an aspect of the present invention;

Figure 3A is a block diagram illustrating the details of an embodiment of an edge router implemented according to an aspect of the present invention;

Figure 3B is a block diagram the details of an alternative embodiment of an edge router implemented according to an aspect of the present invention; and

Figure 4 is a block diagram illustrating the details of an embodiment of a device implemented substantially in the form of software according to an aspect of the present invention.

Detailed Description of the Preferred Embodiments

1. Overview and Discussion of the Invention

In accordance with the present invention, multiple layer-2 virtual circuits are associated with single layer-3 route, and the traffic load of the layer-3 route is distributed on the layer-2 virtual circuits. Thus, a high aggregate bandwidth may be available between two layer-3 devices at either end of a layer-3 route. In addition, the solutions may be implemented without adding any (or at least substantial) complexity to the layer-3 routing protocols.

Several aspects of the invention are described below with reference to example environments for illustration. It should be understood that numerous specific details, relationships, and methods are set forth to provide a full understanding of the invention. One skilled in the relevant art, however, will readily recognize that the invention can be practiced without one or more of the specific details, or with other methods, etc. In other instances, well-known structures or operations are not shown in detail to avoid obscuring the invention.

2. Example Environment

Figure 1 is a block diagram illustrating an example environment in which the present invention can be implemented. The environment is shown containing user systems 110-A, 110-B, 170-A and 170-B, edge routers 120 and 160, and switches 130-A, 130-B, 140-A and 140-B. The switches are shown in ATM backbone 150. Each component is described below in further detail.

The environment is shown containing a few representative components only for illustration. In reality, each environment typically contains many more components. In addition, for conciseness and

clarity, the invention is described with reference to edge router 120 only. However, several aspects are applicable to edge router 160 as well.

User systems 110-A, 110-B communicate with user (or target) systems 170-A and 170-B using ATM backbone 150. Each user system (e.g., 110-A) interfaces with the connected (e.g., user system 110-A is shown connected to edge router 120) edge router(s) using a layer 3 protocol such as Internet Protocol (IP). Each user system may correspond to a computer system or workstation, and can be implemented in a known way.

ATM backbone 150 is shown containing switches 130-A, 130-B, 140-A and 140-B, which provide different physical paths between edge routers 120 and 160. The switches operate consistent with the ATM protocol, and may be implemented in a known way. In general, switches enable edge routers 120 and 160 to communicate with each other using ATM protocol.

It should be understood that ATM is example of a layer-2 protocol and the present invention can be implemented using other packet-based layer-2 protocols such as Frame Relay. Such other implementations are also contemplated to be within the scope and spirit of the present invention.

Edge router 120 interfaces with user systems 110-A and 110-B using IP protocol (an example of a layer-3 protocol), and with switch 130 using ATM (layer-2 protocol). In accordance with the present invention, edge router 120 may use several layer-2 virtual circuits to communicate on a single IP (layer-3) route to edge router 160, and balance the traffic load across the virtual circuits. The manner in which such a benefit may be attained is described below with examples.

3. Method

Figure 2 is a flow-chart illustrating a method in accordance with the present invention. The method is described with reference to Figure 1 for illustration only. However, the method can be implemented in other environments also, and such implementations are contemplated to be within the scope and spirit of the present invention. The method starts in step 201, in which control immediately passes to step 210.

In step 210, a plurality of virtual circuits (e.g., PVCs) are provisioned associated with a single layer-3 route. Thus, with reference to Figure 1, a first virtual circuit may be provisioned on a path formed by switches 130-A and 140-A, and a second virtual circuit may be provisioned on a path formed by switches 130-B and 140-B. The two virtual circuits may be associated with an IP route between the two edge routers 120 and 160.

In step 230, a plurality of layer 3 (IP) datagrams are received in edge router 120, typically from end systems 110-A and 110-B. In step 250, edge router 120 determines the specific (subset of) layer-3 datagrams which need to forwarded on the layer-3 route (noted in step 210). The determination is generally based on examination of the destination IP address contained in the header of each IP datagram, and may be implemented in a known way.

In step 270, the traffic load from edge router 120 to edge router 160 is distributed on the plurality of virtual circuits associated with the layer-3 route. Load distribution generally entails assigning each datagram to one of the virtual circuits, and transmitting the datagram on the assigned virtual circuit in the form of packets suitable for transmission on the layer-2 network.

Several approaches can be employed in assigning datagrams to the individual virtual circuits. It may be preferable to assign datagrams related to the same flow (typically unique combination of source/destination addresses/ports) to the same virtual circuit. For example, all datagrams with the same destination address may be assigned to the same virtual circuit.

Thus, some of the datagrams may be sent on the first virtual circuit and the remaining datagrams may be sent on the second virtual circuit. In an embodiment, the first and second virtual circuits may be associated with other IP routes also as a designer may wish. Thus, the present invention may be used to distribute the traffic load on a single IP route using multiple associated virtual circuits.

While the embodiments are described with reference to using only a single layer-3 router between edge routers 120 and 160, it should be understood that multiple parallel layer-3 routers may be present between the edge routers, with each layer-3 route being associated with a plurality of virtual circuits as described above. Such implementations are contemplated to be within the scope and spirit of the present invention. The description is continued with reference to an embodiment of edge router 120.

4. Edge Router

Figure 3A is a block diagram illustrating the details of an embodiment of edge router 120 as relevant to several aspects of the present invention. Edge router 120 is shown containing inbound interface 310, forwarding block 320, VC (virtual circuit) determination block 330, segmentation block 340, encapsulator 350, and outbound interface 390. Each component is described below in further detail. For illustration, it will be assumed that two PVCs are provisioned on respective physical paths

{switches 130-A and 130-B} and {switches 140-A and 140-B} between edge routers 120 and 160.

Inbound interface 310 provides the physical, electrical and other protocol interfaces to receive layer-3 (IP) datagrams from user systems 110-A and 110-B. Similarly, outbound interface 390 provides the physical, electrical and protocol interfaces to transmit ATM cells on the two PVCs between edge routers 120 and 160. Inbound interface 310 and outbound interface 390 may be implemented in a known way.

Forwarding block 320 receives an IP datagram from inbound interface 310, and determines an IP route the datagram is to be forwarded on. In an embodiment, the destination IP address is compared against entries in forwarding table 325 to retrieve a route entry. The route entry may contain an outbound interface and an IP address of (an interface of) a device (i.e., edge router 160) at the next hop. Forwarding table 325 may be populated using routing protocols and/or manually in a known way.

VC determination block 330 determines the specific one of the virtual circuits to use in forwarding a layer-3 datagram. As the datagrams to edge router 160 can be transmitted on any one of the two PVCs, VC determination block 330 needs to determine the specific one of the PVCs to use for transmitting each datagram. In an embodiment, the IP address of the edge router is used as an index to retrieve the circuit identifiers (VPI/VCI) of the associated PVCs. One of the two PVCs is then selected for each datagram.

Various approaches and considerations can be used in balancing the traffic load on the two PVCs. It may be desirable to allow the same application flow on the same PVC such that the user

systems need not re-sequence the received data. Accordingly, the datagrams related to the same destination IP address may be mapped to the same virtual circuit. In the alternative, additional fields such as port numbers may also be examined to map a flow to the same PVC.

Segmentation block 340 segments each IP datagram into several ATM payload, and may be implemented in a known way. Encapsulator 350 receives each payload and encapsulates the payload into a corresponding ATM cell. The header for each cell is constructed based on a VPI/VCI received from VC determination block 330. The same VPI/VCI is used to encapsulate all cells of a datagram. Due to the encapsulation, the cells of each datagram are transmitted on a PVC determined by VC determination block 330. The cells are passed to outbound interface 390 for transmission on the corresponding port.

From the above, it may be appreciated that edge router 120 distributes the traffic load among several virtual circuits by associating the virtual circuits to a single IP route. Edge router 160 receives the sequence of cells carrying each datagram, and re-constructs the datagram from the cells in a known way. Datagrams may be transmitted in the reverse direction also similarly.

One disadvantage with the embodiment of Figure 3A is that the IP route lookup and the VC lookup are implemented as two separate actions, and the throughput performance of edge router 120 may be impeded. The two actions may be combined, for example, as described below with reference to an alternative embodiment of edge router 120.

5. Alternative Embodiment

Figure 3B is a block diagram illustrating the details of edge router 120 in an alternative embodiment. Only the differences relative to Figure 3A are described for conciseness. Forwarding block 320 and forwarding table 325 are absent in Figure 3B. In addition, VC determination block 380 and forwarding information base (FIB) 385 respectively replace VC determination block 330 and VC table 335.

As described below, VC determination block 380 may efficiently determine the specific virtual circuit on which to send a datagram by examining FIB 385. The operation of VC determination block 380 can be appreciated by understanding the information present in FIB 385. Accordingly, FIB 385 is described first below.

FIB 385 may be implemented to efficiently map layer-3 information to a layer-2 information directly using a single search/lookup such that the mapping time is minimized. The IP route associated with each destination IP address (e.g., using network address and sub-net mask) may be determined a priori, and one of the associated virtual circuits is selected to transmit the datagrams with the destination IP address. In one implementation, only one virtual circuit is associated with each destination IP address in order to quickly process the datagrams.

Thus, in operation, FIB 385 needs to be examined quickly to determine the layer-2 virtual circuit. In an embodiment, FIB 385 is implemented consistent with Cisco Express Forwarding (CEF) Protocol, which is described in a document entitled, "Cisco Express Forwarding" available from Cisco Systems, Inc. (the intended assignee of the present application) and also on the world-wide web at

URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios112/ios112p/gsr/cef.htm, and is incorporated in its entirety herewith. An M-trie structure, well known in the relevant arts, can be used to implemented FIB 385, with the leaves providing the layer-2 header information.

Thus, VC determination block 380 uses a destination IP address (of a received datagram) and traverses the tree of FIB 385 to determine the VPI/VCI to be used for transmitting the datagram. The VPI/VCI information is passed to encapsulator 350. Encapsulator 350 encapsulates the payload segments provided by segmentation block 340 using the VPI/VCI (and other header information) provided by VC determination block 380. The resulting cells are transmitted using outbound interface 390.

Accordingly, the embodiment(s) of Figure 3B can also be used to transmit layer-3 datagrams on multiple virtual circuits. It should be understood that each feature of the present invention can be implemented in a combination of one or more of hardware, software and firmware. In general, when throughput performance is of primary consideration, the implementation is performed more in hardware (e.g., in the form of an application specific integrated circuit).

When cost is of primary consideration, the implementation is performed more in software (e.g., using a processor executing instructions provided in software/firmware). Cost and performance can be balanced by implementing edge router 120 with a desired mix of hardware, software and/or firmware. An embodiment implemented substantially in software is described below.

6. Software Implementation

Figure 4 is a block diagram illustrating the details of edge router 120 in one embodiment. Edge router 120 is shown containing processing unit 410, random access memory (RAM) 420, storage 430, output interface 460, datagram memory 470, network interface 480 and input interface 490. Each component is described in further detail below.

Output interface 460 provides output signals (e.g., display signals to a display unit, not shown) which can form the basis for a suitable user interface for an administrator to interact with edge router 120. Input interface 490 (e.g., interface with a key-board and/or mouse, not shown) enables an administrator to provide any necessary inputs to edge router 120. Output interface 460 and input interface 490 can be used, for example, to enable a network administrator to specify the specific PVCs to be associated with a layer-3 route.

Network interface 480 enables edge router 120 to send and receive data on communication networks using asynchronous transfer mode (ATM) and layer-3 protocols (e.g., IP, DECnet, and, Vines protocol well known in the relevant arts) edge router 120 may be using. Network interface 480, output interface 460 and input interface 490 can be implemented in a known way.

RAM 420, storage 430, and datagram memory 470 may together be referred to as a memory. RAM 420 receives instructions and data on path 450 from storage 430, and provides the instructions to processing unit 410 for execution. In addition, RAM 420 may be used to implement the tables and data structures described above with reference to Figures 3A and 3B.

Datagram memory 470 stores (queues) cells/datagrams received and/or waiting to be forwarded (or otherwise processed) on different ports. Such memories can be employed in the embodiments of Figure 3A and 3B as well when buffering is required. Secondary memory 430 may contain units such as hard drive 435 and removable storage drive 437. Secondary storage 430 may store the software instructions and data, which enable edge router 120 to provide several features in accordance with the present invention.

Some or all of the data and instructions may be provided on removable storage unit 440, and the data and instructions may be read and provided by removable storage drive 437 to processing unit 410 via RAM 420. Floppy drive, magnetic tape drive, CD-ROM drive, DVD Drive, Flash memory, removable memory chip (PCMCIA Card, EPROM) are examples of such removable storage drive 437.

Processing unit 410 may contain one or more processors. Some of the processors can be general purpose processors which execute instructions provided from RAM 420. Some can be special purpose processors adapted for specific tasks (e.g., for memory/queue management). The special purpose processors may also be provided instructions from RAM 420. In general processing unit 410 reads sequences of instructions from various types of memory medium (including RAM 420, storage 430 and removable storage unit 440), and executes the instructions to provide various features of the present invention.

Embodiments according to Figure 4 can be used to implement the approaches described above with reference to Figures 3A and 3B. Some considerations in software implementations described

below.

7. Considerations in Software Implementations

The approach of Figure 3A (in which an IP destination address is first mapped to an IP route and then the specific VC transmitting the datagram is determined for the route) may be implemented substantially in software in the embodiment of Figure 4. In such an embodiment, an interrupt may be generated when each IP datagram is received.

Processing unit 410 may receive the datagram and place the place the datagram in datagram memory 470. Further processing of the datagram may need to be implemented in the form of processes (as opposed to interrupt handlers) under the control of a scheduler. As is well known, interrupt handlers generally receive a pre-emptive priority over scheduler controlled processes. Thus, for quick processing of datagrams, it is generally desirable that the datagrams be handled by interrupt handlers.

As the approach of Figure 3A requires multiple look-ups, it may be undesirable to implement the processing of IP datagrams in interrupt handlers. Accordingly, the embodiment of Figure 3A may not scale to environments requiring quick processing of a large number of datagrams.

An alternative embodiment of Figure 4 may implement the approach of Figure 3B in which the determination of an ATM header (including the VPI/VCI) can be performed quickly using the tree structures. As the lookup can be performed quickly, the datagrams may be processed in interrupt handlers, and the embodiments may scale to environments requiring quick processing of a large number

of datagrams. The implementation of the logic in the form of interrupt handlers will be apparent to one skilled in relevant arts by reading the disclosure provided herein.

Thus, the present invention can be used to implement edge routers which allow the traffic load on IP routes to be balanced on different virtual circuits. As the virtual circuits can be provisioned on different underlying physical paths, a high aggregate amount of bandwidth can be provided between edge routers without potentially not having to upgrade the bandwidth on the physical links.

8. Conclusion

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.